1. Introduction

The TMS34010 Graphics System Processor (**GSP**) is an advanced 32-bit microprocessor optimized for graphics systems. The GSP is a member of the TMS340 family of computer graphics products from Texas Instruments.

A single TMS34010 provides a cost-effective solution in applications that require efficient data manipulation. The GSP can be configured to serve in either a host-based or a stand-alone environment. Systems based on multiple TMS34010 devices are implemented using special features of the GSP's local and host interfaces.

The TMS34010 is well supported by a full set of hardware and software development tools, including a full-speed emulator, a software simulator, an IBM-PC development board, and a C compiler.

Topics covered in this introductory section include:

1.1 TMS34010 Overview

The TMS34010 combines the best features of general-purpose processors and graphics controllers to create a powerful and flexible Graphics System Processor. Key features of the GSP are its speed, high degree of programmability, and efficient manipulation of hardware-supported data types such as pixels and two-dimensional arrays of pixels.

The GSP's unique memory interface reduces the time needed to perform tasks such as bit alignment and masking. The 32-bit architecture supplies the large blocks of continuously-addressable memory necessary in graphics applications. The use of video RAMs facilitates the design of high-bandwidth frame buffers, circumventing the bottleneck often encountered with conventional DRAMs.

The GSP instruction set includes a full complement of general-purpose instructions as well as graphics functions from which a programmer can construct efficient high-level functions. The instructions support arithmetic and Boolean operators, data moves, conditional jumps, and subroutine calls and returns.

The GSP architecture supports a variety of pixel sizes, frame buffer sizes, and screen sizes. On-chip functions have been carefully selected so that no functions tie the GSP to a particular display resolution. This enhances the portability of graphics software, and allows the GSP to adapt to graphics standards such as CGI/CGM, GKS, NAPLPS, PHIGS, and evolving display and terminal management standards.

1.2 Key Features

- Fully programmable 32-bit general-purpose processor
- 128-megabyte address range
- 160-ns instruction cycle time
- On-chip peripheral functions include:
  - Programmable CRT control (horizontal sync, vertical sync, and blanking)
  - Direct interfacing to conventional DRAMs and multiport video RAMs
  - Automatic CRT display refresh
  - Direct communications with an external (host) processor
- Instruction set includes special graphics functions such as pixel processing, XY addressing, and window clip/hit
- Programmable 1, 2, 4, 8, or 16-bit pixel size with 16 Boolean and 6 arithmetic pixel-processing options
- 30 general-purpose 32-bit registers
- 256-byte LRU on-chip instruction cache
- Dedicated 8/16-bit host-processor interface and HOLD/HLDA interface
- 32-bit and 64-bit integer arithmetic
- High-level language support
- Full line of hardware and software development tools including:
  - C compiler
  - Macro assembler
  - Linker
  - Archiver
  - Software libraries
  - XDS (Extended Development Support) in-circuit emulator
  - Software Development Board (SDB)
  - ROM utility
  - Simulator
- 68-pin PLCC package
- 5-V CMOS technology

1.3 Typical Applications

The TMS34010's 32-bit processing power and its ability to handle complex data structures make it well suited for a variety of applications. These include display systems, imaging systems, mass storage, communications, high-speed controllers, and peripheral processing. The GSP's efficient bit manipulation facilitates demanding tasks such as high-quality, proportionally-spaced text. This capability makes it especially useful in applications such as desktop publishing. In graphics display systems, the GSP provides cost-effective performance for color or black-and-white bit-mapped displays. Table 1-1 lists typical end uses of the GSP.

Table 1-1. Typical Applications of the TMS34010

Computers

- Terminals and CRTs
- Electronic publishing
- Laser printers
- Personal computers
- Printers and plotters
- Engineering workstations
- Copiers
- Document readers
- FAX
- Imaging
- Data processing

Industrial Control

- Robotics
- Process control
- Instrumentation
- Motor control
- Navigation

Telecommunications

- Video phones
- PBX

Consumer Electronics

- Automotive displays
- Information terminals
- Cable TV
- Home control
- Video games

1.4 Architectural Overview

Figure 1-1 illustrates the TMS34010's major internal functions and its interfaces to external devices. The on-chip processor executes both graphics instructions and general-purpose instructions. The GSP is a true 32-bit processor, with 32-bit internal data paths, a 32-bit ALU, and a large address space. Thirty 32-bit general-purpose registers, a 32-bit stack pointer, and a 256-byte instruction cache increase performance. Nonprocessor functions included on the chip include CRT timing, screen refresh, and DRAM refresh. Separate physical interfaces are provided for communicating with a host processor, for providing the video timing signals necessary to control a CRT monitor, and for connecting directly to dynamic RAMs and video RAMs.

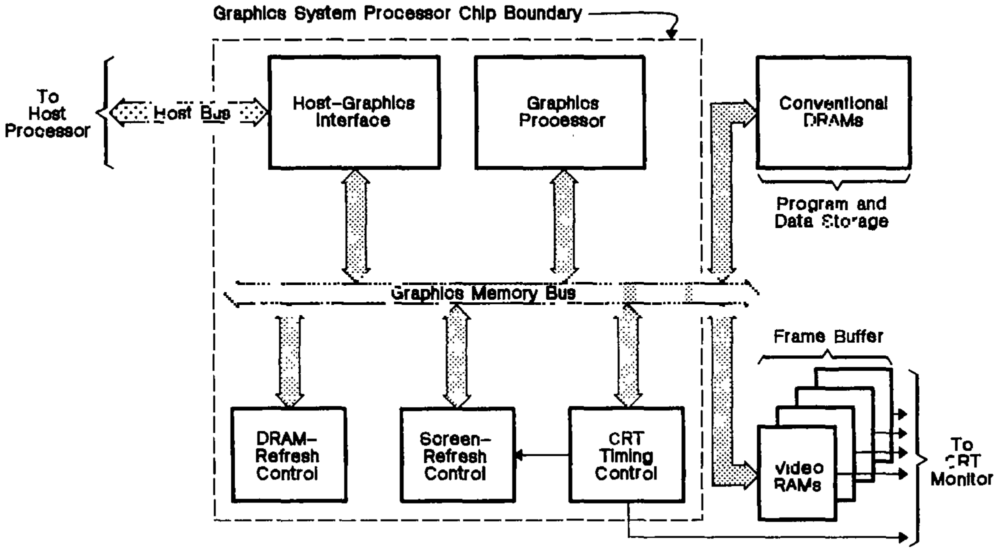


Figure 1-1. System Block Diagram

1.4.1 Other Special Processing Hardware

The TMS34010 CPU functions include the following special processing hardware:

- Hardware for detecting whether a pixel lies within a specified display window.
- Hardware for detecting the leftmost one in a 32-bit register.
- Hardware for expanding a black-and-white pattern to a variable pixel-depth pattern.

1.4.2 TMS34010 Block Diagram

Figure 1-2 illustrates the internal architecture of the TMS34010. The list that follows describes the individual blocks shown in Figure 1-2.

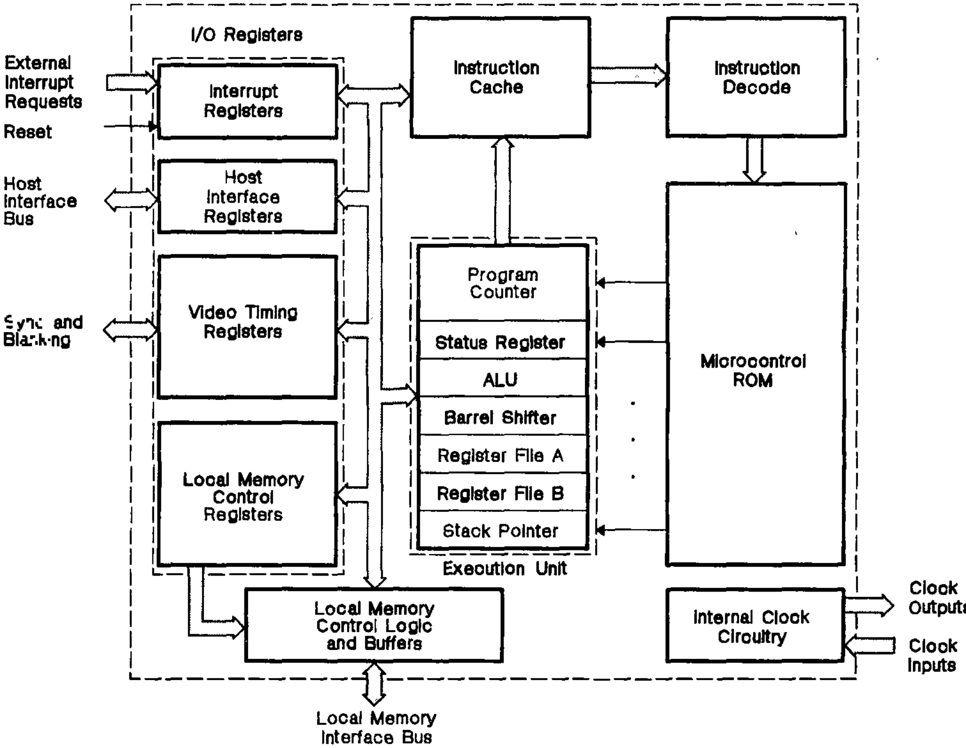


Figure 1-2. Internal Architecture Block Diagram

- **CPU Internal Functions**

  The main internal functions of the TMS34010 are shown in the center of Figure 1-2. Section 5 discusses the CPU registers in detail.

  – The 32-bit **program counter** (PC) points to the next instruction word to be fetched. The PC's four LSBs are always 0.

  – The 32-bit **status register** (ST) specifies the status of the TMS34010 processor. It contains the sign, carry, zero, overflow, interrupt enable, and PixBlt execution status bits. It also specifies the lengths and field extension modes of Fields 0 and 1.

  – **Register files A and B** each contain 15 general-purpose registers, A0–A14 and B0–B14, respectively. The B-file registers are also used as implied operands for the graphics instructions.

The general-purpose register files are dual ported to support parallel data movement. Two separate internal buses route data from the registers to the ALU, and a third bus routes results back to the registers.

- The **stack pointer**, or SP, is available to instructions that operate on either register file.
- The 32-bit **barrel shifter** shifts or rotates 32-bit operands from 1 to 32 bit positions in a single machine state.
- The 32-bit **ALU** is connected to the other CPU components by 32-bit data paths. This allows most register-to-register operations to be performed in a single machine state. (Accessing external memory requires a minimum of two states.) The following actions occur in parallel during a single state:

  1) Two operands are transferred from the selected general-purpose register file to the ALU.
  2) The ALU performs the specified operation on the operands.
  3) The result is routed back to the general-purpose register file.

- **Instruction Cache**

  The TMS34010 contains a 256-byte instruction cache. The cache can contain up to 128 instruction words (an instruction word may be an entire single-word instruction or 16 bits of a multiple-word instruction). Section 5.3 describes instruction cache operation.

- **I/O Registers**

  The TMS34010 has 28 16-bit I/O registers on chip which are dedicated to peripheral control functions. Section 6 provides individual descriptions of each I/O register. The I/O registers can be divided into four categories:

  - Seven **local memory interface registers** are dedicated to memory interface control and configure the memory controller.
  - Fourteen **video timing and screen refresh registers** generate the sync and blanking signals used to drive a CRT, and schedule screen-refresh cycles.
  - Five **host interface registers** are accessible to external host processors as well as to the TMS34010. Status information can be communicated directly through these registers. Large blocks of data in GSP memory can be accessed indirectly through pointer registers.
  - Two **interrupt control registers** provide status information about interrupt requests.

- **Microcontrol ROM**

  The TMS34010 transfers decoded instructions to the microcontrol ROM for interpretation. The microcontrol ROM has 166 control outputs and 808 microstates.

- **Clock Timing Logic**

  The clock timing logic converts the clock input signals to internal timing signals and generates the clock output signals, LCLK1 and LCLK2, used by external devices.

1.5 Manual Organization

The *TMS34010 User's Guide* describes GSP operation, focusing on the GSP's role in applications that involve CRT-based, bit-mapped, graphics systems. The User's Guide is divided into four major sections:

1) General information (Section 1)
2) Architecture (Sections 2–8)
3) Timing (Sections 9–11)
4) Instruction set (Sections 7, 12, and 13)

An extensive index and two reference cards are also provided.

Section 1 Introduction

Provides an overview of the TMS34010, including key features and typical applications. Provides a general overview of the TMS34010 architecture; includes a block diagram and a detailed list describing the elements in the diagram. Discusses manual organization and lists suggested reading.

Section 2 Pin Functions

Illustrates the TMS34010 pinout and contains general pin descriptions. Also describes specific pin functions regarding the host interface, the local bus interface, video timing signals, hold and emulator interface pins, and power, ground, and reset pins.

Section 3 Memory Organization

Discusses 32-bit addressing schemes, the TMS34010 memory map, and the stack.

Section 4 Hardware-Supported Data Structures

Discusses hardware-supported data structures such as fields and pixels. XY addressing is also discussed in this section.

Section 5 CPU Registers and Instruction Cache

Describes general-purpose register files A and B, the status register, the program counter, and the instruction cache.

Section 6 I/O Registers

Provides a detailed discussion of host interface registers, memory-interface control registers, video timing and screen refresh registers, interrupt interface registers, and I/O register addressing. Full-page descriptions of each I/O register are presented alphabetically.

Section 7 Graphics Operations

Discusses graphics instructions such as PixBlts, PIXTs, and related topics such as two-dimensional arrays of pixels, window checking, XY-to-linear conversion, and plane masking.

Section 8 Interrupts, Traps, and Reset

Describes external and internal interrupts, interrupt processing, and reset.

Section 9 Screen Refresh and Video Timing

Describes the horizontal sync, vertical sync, and blanking signals, horizontal and vertical timing, and video RAM control.

Section 10 Host Interface Bus

Discusses host interface pins, registers, and timing.

Section 11 Local Memory Interface Bus

Discusses local memory interface timing, addressing mechanisms, and data manipulation at the local memory interface.

Section 12 Assembly Language Instruction Set

Discusses addressing modes, summarizes move, PIXBLT, and PIXT instruction variations, and presents the entire TMS34010 assembly language instruction set in alphabetical order.

Section 13 Instruction Timings

Contains an overview of timing for general instructions, and specific timing information for move and graphics instructions.

Appendix A TMS34010 Data Sheet

Appendix B Emulation Guidelines for Prototyping

Appendix C Software Compatibility with Future GSPs

Appendix D Glossary

1.6 References and Suggested Reading

The following books and articles provide further background in graphics and system concepts associated with graphics.

Artwick, Bruce A. *Applied Concepts in Microcomputer Graphics*. Englewood Cliffs, New Jersey: Prentice-Hall, 1984.

Bresenham, J.E. "Algorithm for Computer Control of a Digital Plotter." *IBM Systems Journal* 4 No.1 (1965): 25–30.

Bresenham, J.E. "A Linear Algorithm for Incremental Display of Digital Arcs." *Communications of the ACM* 20 (Feb. 1977): 100–106.

Cody, William J. Jr., and William Waite. *Software Manual for the Elementary Functions*. Englewood Cliffs, New Jersey: Prentice-Hall, 1980.

Foley, James, and Andries van Dam. *Fundamentals of Interactive Computer Graphics*. Reading, Massachussetts: Addison-Wesley, 1982.

Gupta, Satish. "Architectures and Algorithms for Parallel Updates of Raster Scan Displays." Tech. Report CMU-CS-82-111, Computer Science Dept., Carnegie Mellon University, 1981.

Ingalls, D.H. "The Smalltalk Graphics Kernel." Special issue on Smalltalk, *Byte*, August 1981, pp. 168-194.

Kernighan, B., and D. Ritchie *The "C" Programming Language*. Englewood Cliffs, New Jersey: Prentice-Hall, 1978.

Kochan, Stephen G. *Programming in C*. Hasbrouck Heights, New Jersey: Hayden Book Company, 1983.

Newman, W.M., and R.F. Sproull. *Principles of Interactive Computer Graphics*. 2nd ed. New York: McGraw-Hill, 1979.

Pike, Rob. "Graphics in Overlapping Bitmap Layers." *ACM Transactions On Graphics* 2 (April 1983): 135–160.

Pitteway, M.L.V. "Algorithm for Drawing Ellipses or Hyperbolae with a Digital Plotter." *Computer Journal* 10 (Nov. 1967): 24–35.

Porter, T. and T. Duff. "Composing Digital Images." *Computer Graphics*, July 1984, pp. 253-259.

Sproull, R.F. and I.E. Sutherland. "A Clipping Divider." *Fall Joint Computer Conference* Washington, DC: Thompson Books, 1968.

Van Aken, Jerry R. "An Efficient Ellipse-Drawing Algorithm." *IEEE Computer Graphics & Applications* 4 (Sept. 1984): 24–35.